| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/XAResource.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/transaction/xa/XAException.html)   [**NEXT CLASS**](http://docs.google.com/javax/transaction/xa/Xid.html) | [**FRAMES**](http://docs.google.com/index.html?javax/transaction/xa/XAResource.html)    [**NO FRAMES**](http://docs.google.com/XAResource.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | CONSTR | [METHOD](#2et92p0) | DETAIL: [FIELD](#tyjcwt) | CONSTR | [METHOD](#2jxsxqh) |

## **javax.transaction.xa**

Interface XAResource

public interface **XAResource**

The XAResource interface is a Java mapping of the industry standard XA interface based on the X/Open CAE Specification (Distributed Transaction Processing: The XA Specification).

The XA interface defines the contract between a Resource Manager and a Transaction Manager in a distributed transaction processing (DTP) environment. A JDBC driver or a JMS provider implements this interface to support the association between a global transaction and a database or message service connection.

The XAResource interface can be supported by any transactional resource that is intended to be used by application programs in an environment where transactions are controlled by an external transaction manager. An example of such a resource is a database management system. An application may access data through multiple database connections. Each database connection is enlisted with the transaction manager as a transactional resource. The transaction manager obtains an XAResource for each connection participating in a global transaction. The transaction manager uses the start method to associate the global transaction with the resource, and it uses the end method to disassociate the transaction from the resource. The resource manager is responsible for associating the global transaction to all work performed on its data between the start and end method invocations.

At transaction commit time, the resource managers are informed by the transaction manager to prepare, commit, or rollback a transaction according to the two-phase commit protocol.

| **Field Summary** | |
| --- | --- |
| static int | [**TMENDRSCAN**](http://docs.google.com/javax/transaction/xa/XAResource.html#TMENDRSCAN)            Ends a recovery scan. |
| static int | [**TMFAIL**](http://docs.google.com/javax/transaction/xa/XAResource.html#TMFAIL)            Disassociates the caller and marks the transaction branch rollback-only. |
| static int | [**TMJOIN**](http://docs.google.com/javax/transaction/xa/XAResource.html#TMJOIN)            Caller is joining existing transaction branch. |
| static int | [**TMNOFLAGS**](http://docs.google.com/javax/transaction/xa/XAResource.html#TMNOFLAGS)            Use TMNOFLAGS to indicate no flags value is selected. |
| static int | [**TMONEPHASE**](http://docs.google.com/javax/transaction/xa/XAResource.html#TMONEPHASE)            Caller is using one-phase optimization. |
| static int | [**TMRESUME**](http://docs.google.com/javax/transaction/xa/XAResource.html#TMRESUME)            Caller is resuming association with a suspended transaction branch. |
| static int | [**TMSTARTRSCAN**](http://docs.google.com/javax/transaction/xa/XAResource.html#TMSTARTRSCAN)            Starts a recovery scan. |
| static int | [**TMSUCCESS**](http://docs.google.com/javax/transaction/xa/XAResource.html#TMSUCCESS)            Disassociates caller from a transaction branch. |
| static int | [**TMSUSPEND**](http://docs.google.com/javax/transaction/xa/XAResource.html#TMSUSPEND)            Caller is suspending (not ending) its association with a transaction branch. |
| static int | [**XA\_OK**](http://docs.google.com/javax/transaction/xa/XAResource.html#XA_OK)            The transaction work has been prepared normally. |
| static int | [**XA\_RDONLY**](http://docs.google.com/javax/transaction/xa/XAResource.html#XA_RDONLY)            The transaction branch has been read-only and has been committed. |

| **Method Summary** | |
| --- | --- |
| void | [**commit**](http://docs.google.com/javax/transaction/xa/XAResource.html#commit(javax.transaction.xa.Xid,%20boolean))([Xid](http://docs.google.com/javax/transaction/xa/Xid.html) xid, boolean onePhase)            Commits the global transaction specified by xid. |
| void | [**end**](http://docs.google.com/javax/transaction/xa/XAResource.html#end(javax.transaction.xa.Xid,%20int))([Xid](http://docs.google.com/javax/transaction/xa/Xid.html) xid, int flags)            Ends the work performed on behalf of a transaction branch. |
| void | [**forget**](http://docs.google.com/javax/transaction/xa/XAResource.html#forget(javax.transaction.xa.Xid))([Xid](http://docs.google.com/javax/transaction/xa/Xid.html) xid)            Tells the resource manager to forget about a heuristically completed transaction branch. |
| int | [**getTransactionTimeout**](http://docs.google.com/javax/transaction/xa/XAResource.html#getTransactionTimeout())()            Obtains the current transaction timeout value set for this XAResource instance. |
| boolean | [**isSameRM**](http://docs.google.com/javax/transaction/xa/XAResource.html#isSameRM(javax.transaction.xa.XAResource))([XAResource](http://docs.google.com/javax/transaction/xa/XAResource.html) xares)            This method is called to determine if the resource manager instance represented by the target object is the same as the resouce manager instance represented by the parameter *xares*. |
| int | [**prepare**](http://docs.google.com/javax/transaction/xa/XAResource.html#prepare(javax.transaction.xa.Xid))([Xid](http://docs.google.com/javax/transaction/xa/Xid.html) xid)            Ask the resource manager to prepare for a transaction commit of the transaction specified in xid. |
| [Xid](http://docs.google.com/javax/transaction/xa/Xid.html)[] | [**recover**](http://docs.google.com/javax/transaction/xa/XAResource.html#recover(int))(int flag)            Obtains a list of prepared transaction branches from a resource manager. |
| void | [**rollback**](http://docs.google.com/javax/transaction/xa/XAResource.html#rollback(javax.transaction.xa.Xid))([Xid](http://docs.google.com/javax/transaction/xa/Xid.html) xid)            Informs the resource manager to roll back work done on behalf of a transaction branch. |
| boolean | [**setTransactionTimeout**](http://docs.google.com/javax/transaction/xa/XAResource.html#setTransactionTimeout(int))(int seconds)            Sets the current transaction timeout value for this XAResource instance. |
| void | [**start**](http://docs.google.com/javax/transaction/xa/XAResource.html#start(javax.transaction.xa.Xid,%20int))([Xid](http://docs.google.com/javax/transaction/xa/Xid.html) xid, int flags)            Starts work on behalf of a transaction branch specified in xid. |

| **Field Detail** |
| --- |

### TMENDRSCAN

static final int **TMENDRSCAN**

Ends a recovery scan.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.transaction.xa.XAResource.TMENDRSCAN)

### TMFAIL

static final int **TMFAIL**

Disassociates the caller and marks the transaction branch rollback-only.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.transaction.xa.XAResource.TMFAIL)

### TMJOIN

static final int **TMJOIN**

Caller is joining existing transaction branch.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.transaction.xa.XAResource.TMJOIN)

### TMNOFLAGS

static final int **TMNOFLAGS**

Use TMNOFLAGS to indicate no flags value is selected.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.transaction.xa.XAResource.TMNOFLAGS)

### TMONEPHASE

static final int **TMONEPHASE**

Caller is using one-phase optimization.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.transaction.xa.XAResource.TMONEPHASE)

### TMRESUME

static final int **TMRESUME**

Caller is resuming association with a suspended transaction branch.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.transaction.xa.XAResource.TMRESUME)

### TMSTARTRSCAN

static final int **TMSTARTRSCAN**

Starts a recovery scan.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.transaction.xa.XAResource.TMSTARTRSCAN)

### TMSUCCESS

static final int **TMSUCCESS**

Disassociates caller from a transaction branch.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.transaction.xa.XAResource.TMSUCCESS)

### TMSUSPEND

static final int **TMSUSPEND**

Caller is suspending (not ending) its association with a transaction branch.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.transaction.xa.XAResource.TMSUSPEND)

### XA\_RDONLY

static final int **XA\_RDONLY**

The transaction branch has been read-only and has been committed.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.transaction.xa.XAResource.XA_RDONLY)

### XA\_OK

static final int **XA\_OK**

The transaction work has been prepared normally.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.transaction.xa.XAResource.XA_OK)

| **Method Detail** |
| --- |

### commit

void **commit**([Xid](http://docs.google.com/javax/transaction/xa/Xid.html) xid,  
 boolean onePhase)  
 throws [XAException](http://docs.google.com/javax/transaction/xa/XAException.html)

Commits the global transaction specified by xid.

**Parameters:**xid - A global transaction identifieronePhase - If true, the resource manager should use a one-phase commit protocol to commit the work done on behalf of xid. **Throws:** [XAException](http://docs.google.com/javax/transaction/xa/XAException.html) - An error has occurred. Possible XAExceptions are XA\_HEURHAZ, XA\_HEURCOM, XA\_HEURRB, XA\_HEURMIX, XAER\_RMERR, XAER\_RMFAIL, XAER\_NOTA, XAER\_INVAL, or XAER\_PROTO.

If the resource manager did not commit the transaction and the paramether onePhase is set to true, the resource manager may throw one of the XA\_RB\* exceptions. Upon return, the resource manager has rolled back the branch's work and has released all held resources.

### end

void **end**([Xid](http://docs.google.com/javax/transaction/xa/Xid.html) xid,  
 int flags)  
 throws [XAException](http://docs.google.com/javax/transaction/xa/XAException.html)

Ends the work performed on behalf of a transaction branch. The resource manager disassociates the XA resource from the transaction branch specified and lets the transaction complete.

If TMSUSPEND is specified in the flags, the transaction branch is temporarily suspended in an incomplete state. The transaction context is in a suspended state and must be resumed via the start method with TMRESUME specified.

If TMFAIL is specified, the portion of work has failed. The resource manager may mark the transaction as rollback-only

If TMSUCCESS is specified, the portion of work has completed successfully.

**Parameters:**xid - A global transaction identifier that is the same as the identifier used previously in the start method.flags - One of TMSUCCESS, TMFAIL, or TMSUSPEND. **Throws:** [XAException](http://docs.google.com/javax/transaction/xa/XAException.html) - An error has occurred. Possible XAException values are XAER\_RMERR, XAER\_RMFAILED, XAER\_NOTA, XAER\_INVAL, XAER\_PROTO, or XA\_RB\*.

### forget

void **forget**([Xid](http://docs.google.com/javax/transaction/xa/Xid.html) xid)  
 throws [XAException](http://docs.google.com/javax/transaction/xa/XAException.html)

Tells the resource manager to forget about a heuristically completed transaction branch.

**Parameters:**xid - A global transaction identifier. **Throws:** [XAException](http://docs.google.com/javax/transaction/xa/XAException.html) - An error has occurred. Possible exception values are XAER\_RMERR, XAER\_RMFAIL, XAER\_NOTA, XAER\_INVAL, or XAER\_PROTO.

### getTransactionTimeout

int **getTransactionTimeout**()  
 throws [XAException](http://docs.google.com/javax/transaction/xa/XAException.html)

Obtains the current transaction timeout value set for this XAResource instance. If XAResource.setTransactionTimeout was not used prior to invoking this method, the return value is the default timeout set for the resource manager; otherwise, the value used in the previous setTransactionTimeout call is returned.

**Returns:**the transaction timeout value in seconds. **Throws:** [XAException](http://docs.google.com/javax/transaction/xa/XAException.html) - An error has occurred. Possible exception values are XAER\_RMERR and XAER\_RMFAIL.

### isSameRM

boolean **isSameRM**([XAResource](http://docs.google.com/javax/transaction/xa/XAResource.html) xares)  
 throws [XAException](http://docs.google.com/javax/transaction/xa/XAException.html)

This method is called to determine if the resource manager instance represented by the target object is the same as the resouce manager instance represented by the parameter *xares*.

**Parameters:**xares - An XAResource object whose resource manager instance is to be compared with the resource manager instance of the target object. **Returns:***true* if it's the same RM instance; otherwise *false*. **Throws:** [XAException](http://docs.google.com/javax/transaction/xa/XAException.html) - An error has occurred. Possible exception values are XAER\_RMERR and XAER\_RMFAIL.

### prepare

int **prepare**([Xid](http://docs.google.com/javax/transaction/xa/Xid.html) xid)  
 throws [XAException](http://docs.google.com/javax/transaction/xa/XAException.html)

Ask the resource manager to prepare for a transaction commit of the transaction specified in xid.

**Parameters:**xid - A global transaction identifier. **Returns:**A value indicating the resource manager's vote on the outcome of the transaction. The possible values are: XA\_RDONLY or XA\_OK. If the resource manager wants to roll back the transaction, it should do so by raising an appropriate XAException in the prepare method. **Throws:** [XAException](http://docs.google.com/javax/transaction/xa/XAException.html) - An error has occurred. Possible exception values are: XA\_RB\*, XAER\_RMERR, XAER\_RMFAIL, XAER\_NOTA, XAER\_INVAL, or XAER\_PROTO.

### recover

[Xid](http://docs.google.com/javax/transaction/xa/Xid.html)[] **recover**(int flag)  
 throws [XAException](http://docs.google.com/javax/transaction/xa/XAException.html)

Obtains a list of prepared transaction branches from a resource manager. The transaction manager calls this method during recovery to obtain the list of transaction branches that are currently in prepared or heuristically completed states.

**Parameters:**flag - One of TMSTARTRSCAN, TMENDRSCAN, TMNOFLAGS. TMNOFLAGS must be used when no other flags are set in the parameter. **Returns:**The resource manager returns zero or more XIDs of the transaction branches that are currently in a prepared or heuristically completed state. If an error occurs during the operation, the resource manager should throw the appropriate XAException. **Throws:** [XAException](http://docs.google.com/javax/transaction/xa/XAException.html) - An error has occurred. Possible values are XAER\_RMERR, XAER\_RMFAIL, XAER\_INVAL, and XAER\_PROTO.

### rollback

void **rollback**([Xid](http://docs.google.com/javax/transaction/xa/Xid.html) xid)  
 throws [XAException](http://docs.google.com/javax/transaction/xa/XAException.html)

Informs the resource manager to roll back work done on behalf of a transaction branch.

**Parameters:**xid - A global transaction identifier. **Throws:** [XAException](http://docs.google.com/javax/transaction/xa/XAException.html) - An error has occurred.

### setTransactionTimeout

boolean **setTransactionTimeout**(int seconds)  
 throws [XAException](http://docs.google.com/javax/transaction/xa/XAException.html)

Sets the current transaction timeout value for this XAResource instance. Once set, this timeout value is effective until setTransactionTimeout is invoked again with a different value. To reset the timeout value to the default value used by the resource manager, set the value to zero. If the timeout operation is performed successfully, the method returns *true*; otherwise *false*. If a resource manager does not support explicitly setting the transaction timeout value, this method returns *false*.

**Parameters:**seconds - The transaction timeout value in seconds. **Returns:***true* if the transaction timeout value is set successfully; otherwise *false*. **Throws:** [XAException](http://docs.google.com/javax/transaction/xa/XAException.html) - An error has occurred. Possible exception values are XAER\_RMERR, XAER\_RMFAIL, or XAER\_INVAL.

### start

void **start**([Xid](http://docs.google.com/javax/transaction/xa/Xid.html) xid,  
 int flags)  
 throws [XAException](http://docs.google.com/javax/transaction/xa/XAException.html)

Starts work on behalf of a transaction branch specified in xid. If TMJOIN is specified, the start applies to joining a transaction previously seen by the resource manager. If TMRESUME is specified, the start applies to resuming a suspended transaction specified in the parameter xid. If neither TMJOIN nor TMRESUME is specified and the transaction specified by xid has previously been seen by the resource manager, the resource manager throws the XAException exception with XAER\_DUPID error code.

**Parameters:**xid - A global transaction identifier to be associated with the resource.flags - One of TMNOFLAGS, TMJOIN, or TMRESUME. **Throws:** [XAException](http://docs.google.com/javax/transaction/xa/XAException.html) - An error has occurred. Possible exceptions are XA\_RB\*, XAER\_RMERR, XAER\_RMFAIL, XAER\_DUPID, XAER\_OUTSIDE, XAER\_NOTA, XAER\_INVAL, or XAER\_PROTO.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/XAResource.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/transaction/xa/XAException.html)   [**NEXT CLASS**](http://docs.google.com/javax/transaction/xa/Xid.html) | [**FRAMES**](http://docs.google.com/index.html?javax/transaction/xa/XAResource.html)    [**NO FRAMES**](http://docs.google.com/XAResource.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | CONSTR | [METHOD](#2et92p0) | DETAIL: [FIELD](#tyjcwt) | CONSTR | [METHOD](#2jxsxqh) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).